Carnatic Raga Identifier using Machine Learning and Deep Learning Algorithms

Govinda Madhava B S

[#]*Computer Science Department, PES University*
100 Feet Ring Road, Banashankari Stage III, Dwaraka Nagar,
Banashankari, Bengaluru, Karnataka 560085

Abstract— **With the availability of a lot of Carnatic Music files (audio and video) on the internet today, classifying the raga a particular audio file belongs to is a challenge. Using the appropriate Machine Learning and Deep Learning algorithms and the appropriate training and validation datasets, it is possible to classify the audio snippet into a particular raga with a good rate of success. This report compares a few common Machine Learning and Deep Learning algorithms, in their rates of success in predicting the raga a given test audio file belongs to.**

***Keywords*— [6] Carnatic Music, [7] ragas, Machine Learning, Deep Learning**

I. Introduction and Motivation

Listening to a given audio/music recording and identifying the Carnatic Raga it belongs to, is part of the routine for a music student. It takes immense amounts of practice, identifying and remembering the unique characteristics of each raga, being able to distinguish one raga from another and experience, to successfully identify the raga an audio belongs to.

Being a student of Carnatic Music for the past 15+ years, and after getting introduced to the fields of ML and DL through Engineering, it was only intuitive for me to try to implement the idea of training machine to identify the ragas. I thought it would be a good challenge and more importantly, a great learning opportunity for me to test my skills, gain knowledge and hopefully, help many other music students in their journey!

II. References and Related Research

1. The YouTube lecture series by “Valerio Velardo” titled “Audio Signal Processing for Machine Learning” was very helpful in introducing the basics and concepts of Audio Signal Processing for ML and working with audio files, features in python. [1]
2. Music Information Retrieval: Raga Identification using Machine Learning (IIT Madras) [2] and [3]
3. Raga Identification using ML [4]
4. Identifying Ragas in Carnatic Music with Machine Learning by Sridhar Ravikoti [5]

III. Approach

Considering there are 1000+ identifiable ragas in Carnatic Music, I have only considered 2 ragas i.e., Thodi and Kambodi. The objective of this program is to classify a given test audio as belonging to either of the above mentioned two ragas (for purpose of this project, during the prediction phase, test audio strictly belonging to either raga only is chosen).

Data Collection: To train the models, I have chosen about 10 recordings consisting a mixture of vocal (female and male) and instrumental audio files.

Data Preparation: All files are converted to wav format so that they can be loaded using librosa module of python. The user-defined function converts the training audio files into snippets of 10 seconds, with the specified jump.

Raw signal: The audio which gets loaded via librosa module is a time series of raw amplitude.

Fourier transform and spectrogram: Since a raga is a constrained play of relative frequencies, intuition says it would be better to have our features in the frequency domain. Fourier transform of the audio series in time domain gets us the frequencies and their magnitudes in that particular timeframe.

MFCCs: Although MFCCs were helpful in other audio processing applications like genre detection, speech recognition, etc it was not helpful for our current project. MFCCs for both the considered ragas is not distinguishable. Models could not be trained satisfactorily.

Main Frequency: The above three feature extraction techniques have been previously used in many studies/papers but did not yield satisfactory results for this case study. Armed with my modest knowledge of Raga and sound theory, I decided to explore new features.

Main-Frequency Counts: I tried a new feature which lists the counts of the main frequencies for the list of all available frequencies after the fourier transform.

IV. DNN Model Tuning

In deep learning, we have the concept of loss, which tells us how poorly the model is performing at that current instant. Now we need to use this loss to train our network such that it performs better. Essentially what we need to do is to take the loss and try to minimize it, because a lower loss means our model is going to perform better. The process of minimizing (or maximizing) any mathematical expression is called optimization.

Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate to reduce the losses. Optimizers are used to solve optimization problems by minimizing the function.

Tuning of each model explained in detail below in section VII.

V. Machine learning and deep learning models used

1. Linear Regression (LR) [8]
2. Support Vector Machine Algorithm (SVM) [9]
3. Extreme Gradient Boosting (XGBoost) [10]
4. Deep Neural Network (DNN) [11]
5. Long short-term memory algorithms (LSTMs) [12]

VI. Unique contribution

I have compared the accuracies of the following ML models:

- Logistic Regression
- SVM (rbf)
- XGBoost

DL Models:

- DNNs
- LSTMs

Apart from trying to combine Deep Learning techniques along with the Machine Learning algorithms to obtain higher accuracy in predicting the ragas of test audio files, I have contributed my own audio files to assist in training the models.

VII. System architecture, design and implementation

I have created 4 folders (thodi_test, kambodi_test, thodi_training and kambodi_training) which contain the required audio files to train and test the models.

Different variations of the below parameters were tried trial-and-error method) to get the optimal output:

Optimizer (SGD/Adam): SGD solved the Gra Descent problem by using only single record updates parameters. But, still, SGD is slo converge because it needs forward and backward propagation for every record. And the path to reach global minima becomes very noisy.

Learning Rate (0.01 to 0.00001): In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.

Batch size (8 to 128): Batch size is a term used in machine learning and refers to the number of training examples utilized in one iteration.

No. of deep layers (2 to 6): Traditionally, neural networks only had three types of layers: hidden, input and output. These are all really the same type of layer if you just consider that input layers are fed from external data (not a previous layer) and output feed data to an external destination (not the next layer).

Kernel initializers (RandomUniform/HeUnifrom): Initializers define the way to set the initial random weights of Keras layers.

Regularization (L1/L2): Regularization is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting. The commonly used regularization techniques are: L1 regularization, L2 regularization, dropout regularisation

Dropout layers: The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. Inputs not set to 0 are scaled up by 1/(1 - rate) such that the sum over all inputs is unchanged.

Analysing the results

It is observed that there is overfitting of the model. The model doesn't fare well on the test set as compared to validation set but the predictions are still satisfactory considering the dataset used and the amount of training data available.

Final set of predictions for unknown 16 audio files:

5	Thodi	Thodi
6	Thodi	Kambodi
7	Thodi	Thodi
8	Thodi	Thodi
9	Kambodi	Kambodi
10	Kambodi	Thodi
11	Kambodi	Kambodi
12	Kambodi	Kambodi
13	Kambodi	Kambodi
14	Kambodi	Thodi
15	Kambodi	Kambodi
16	Kambodi	Kambodi

IX. CONCLUSION

From the above results, we observe that among the ML classifiers: LR(80%) > XGB (~65%) > SVM (~66%)

While using DL, we achieved an accuracy of ~70%

Hence, considering the quality of used training data and test cases used to predict and calculating the accuracy, we have achieved 80% accuracy in predicting the raga for a random audio test file.

With further improvements in the training audio files and a better dataset, using more specific test data, higher accuracies can be achieved.

X. FUTURE WORK

I plan to further expand the scope of the raga identifier, by including more ragas and trying to improve the accuracy too. Furthermore, we can use the same algorithm to use ML and DL algorithms to recognize the voice of the performer.

These can be used as an innovative kind of biometrics to lock/unlock digital devices, which recognize the voice of the authorized user only.

REFERENCES

[1] https://www.youtube.com/playlist?list=PL-wATfeyAMNqIee7cH3q1bh4QJFAaeNv0

[2] https://www.iitm.ac.in/donlab/website_files/thesis/MTech/thesis_padma.pdf

[3] https://publications.iitm.ac.in/publication/classification-of-melodic-motifs-in-raga-music-with-time-series-2

[4] https://medium.com/@shreyas.divan/raga-identification-using-ml-and-dl-a95b51f47044

[5] https://www.linkedin.com/pulse/identifying-ragas-carnatic-music-machine-learning-sridhar-ravikoti/

[6] https://en.wikipedia.org/wiki/Carnatic_music

[7] https://en.wikipedia.org/wiki/Raga

[8] https://en.wikipedia.org/wiki/Logistic_regression

[9] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[10] https://www.geeksforgeeks.org/xgboost/

[11] https://www.sciencedirect.com/topics/engineering/deep-neural-network

[12] https://en.wikipedia.org/wiki/Long_short-term_memory

[13] Link to the GitHub repo: https://github.com/GovindaMadhava/AdvAlgo_Raga

Sr No	Actual Raga/Scale	Predicted Raga/Scale
1	Thodi	Kambodi
2	Thodi	Thodi
3	Thodi	Thodi
4	Thodi	Thodi
5	Thodi	Thodi
6	Thodi	Kambodi
7	Thodi	Thodi
8	Thodi	Thodi
9	Kambodi	Kambodi
10	Kambodi	Thodi
11	Kambodi	Kambodi
12	Kambodi	Kambodi
13	Kambodi	Kambodi
14	Kambodi	Thodi
15	Kambodi	Kambodi
16	Kambodi	Kambodi